Research Statement

Richard Peng

My research is focused on the design of methods that provably solve large problems quickly. Many computational problems induced by practice arise at the intersection of combinatorics, machine learning, optimization, and statistics. Combining insights from these areas often leads to better algorithms for such problems, as well as improvements to these areas themselves. This multifaceted approach of bringing together combinatorial, numerical, and randomized algorithms has proven to be very fruitful in my work, and I plan to continue pursuing it.

During my graduate studies at Carnegie Mellon University (CMU) and my current instructorship at the Massachusetts Institute of Technology (MIT), I worked on algorithmic questions stemming from combinatorial and scientific flow problems. Many of these problems can be solved via systems of linear equations in the Laplacian matrices of undirected graphs. These matrices offer a key connection between algebraic and combinatorial objects that forms the foundation of the Laplacian paradigm of designing efficient algorithms. Work in this direction, which originated as spectral algorithms due to its connection to spectral graph theory, led to progress on many well-studied questions in algorithms. My representative results include:

- The current fastest algorithms for solving linear systems involving graph Laplacians, a core primitive in spectral algorithms and a well-studied routine in scientific computing. These algorithms are based on constructing ultra-sparse approximations to graphs, and representing the inverses of graph Laplacians by a few sparse matrices.

- The first $O(m \text{polylog}(n))$ time algorithm for approximating undirected maximum flows, a fundamental problem in combinatorial optimization that has motivated the development of many algorithmic tools.

- The first nearly-optimal subsampling algorithms that preserve the $\ell_1$-norms of matrix-vector products, a common structure in data analysis. This result showed that after suitable transformations the $\ell_1$-norm behaves analogously to the $\ell_2$-norm, or Euclidean distance, under sampling. It also gave the first efficient algorithm for computing these transformations.

These algorithms are inherently parallel, making frameworks based on them promising starting points for practical routines. Insights from these frameworks often resurface in graph algorithms, data structures, machine learning, and computational geometry, and I’ve also explored these connections. While most of my work has been theoretical, I am highly interested in their practical implications, and have been involved in projects on medical imaging, data mining, scientific computing, and large scale data processing. My current and past collaborators include researchers from the University of Pittsburgh Medical Center, DSO National Laboratories, and Baidu Research.
Current and Future Directions

A more integrated view of graph theoretic, algebraic, and randomized approaches has led to new algorithms and more systematic studies of widely used heuristics. I believe this is a direction that will lead to progress on key questions in many areas. In the near future, I’m interested in pursuing the following directions:

1. The incorporation of combinatorial and randomized methods into numerical analysis, leading to improved algorithms for widely used computational tasks such as solving linear systems, low rank approximations, sparse inverses, finite element simulations, and heat kernels.

2. The design of combinatorial optimization algorithms motivated by algebraic routines, leading to a better understanding of the tradeoffs between combinatorial and numerical approaches and faster algorithms for more classical problems in algorithmic graph theory.

3. The interplay between numerical analysis and statistics, leading to a wider range of computable statistical quantities, new methods of analyzing random processes, and improved $\ell_p$-norm preserving embeddings of subspaces of high dimensional spaces into lower dimensions.

The growing popularity of the Laplacian paradigm in machine learning, graphics, and image processing is indicative of the potential impact of this method of designing algorithms. As practical questions tend to be application-driven, I intend to work closely with collaborators on these topics. Questions that I currently find particularly intriguing are the connections between solving linear systems and sampling from graphical models, warm-start during the iterations of second-order optimization methods, and structure-preserving subsampling for wider classes of objects / distance measures. Of course, the range of questions is much wider here, but my approach towards designing algorithms has consistently led to new insights on such questions in the past.

Laplacian Solvers

My collaborators and I developed the current fastest sequential and parallel Laplacian solvers. Aside from being a key question in algorithm design, practical routines for solving linear systems involving graph Laplacians are widely studied in scientific computing. Algorithms based on such routines, often referred to as spectral algorithms, have widespread applications in clustering, image processing and social network analysis.

Although solving linear systems is one of the most studied algorithmic problems, the current theoretical guarantees for solvers are far from optimal. An important subclass of linear systems that occur frequently in scientific computing is M-matrices, which is algorithmically equivalent to symmetric diagonally dominant matrices and graph Laplacians. Spielman and Teng gave the first nearly-linear time solver for linear systems in graph Laplacians that had a runtime bound of $m \log^{70} n$. This led to the open question of whether such solvers can be made practical. In a sequence of result that spanned my graduate studies at CMU, I gave strong evidence for the feasibility of fast graph Laplacian solvers that also have theoretical guarantees. These algorithms run in time close to $m \log^{1/2} n$, and simplified most parts of the Spielman-Teng framework. Ideas from these constructions motivated subsequent results by Kelner, Lee, Orecchia, Sidford, Zhu that drew a close connection between this framework and accelerated versions of the randomized Kaczmarz method.
Subsequently, Daniel Spielman and I gave the first nearly-linear work, polylog depth solver for linear systems in graph Laplacians. This method has close connections to multigrid methods, a popular method for solving large sparse linear systems in practice. It is based on repeated squaring while keeping all intermediate matrices sparse, and gives a provably sparse approximation of the inverse. More recently, Yu Cheng, Dehua Cheng, Yan Liu, Shang-Hua Teng, and I showed that this method leads to a general framework for numerical routines.

Optimization

The Laplacian paradigm led to a multitude of novel methods for combining combinatorial and numerical routines. Results by Christiano, Daitch, Kelner, Madry, Spielman, and Teng showed that linear system solvers are closely connected to classical problems in combinatorial graph theory such as maximum flow and minimum cost flow. Both of these problems are well studied in combinatorial optimization, and have motivated the development of many key algorithmic techniques.

I gave the first $O(m \text{polylog}(n))$ time algorithm for approximating undirected maximum flows. The result combined algorithms by Kelner, Lee and Orecchia, Räcke, Shah, Sherman, Sidford, and Täubig via ultra-sparsifiers, a central component in fast sequential algorithms for solving linear systems in graph Laplacians. The correspondence between this algorithm and Laplacian solvers suggest a much closer connection between linear system solvers and combinatorial optimization.

This optimization view of problems also supplements scientific computing techniques that form the basis of linear system solvers, leading to extensions of the Laplacian paradigm to matrices with more intricate structures. With Jonathan Kelner and Gary Miller, I initiated the study of solvers for linear systems associated with multicommodity flow problems by modifying both approximate flow algorithms and linear system solvers to meet at a specific class of well-structured linear systems. With Michael Cohen, Brittany Fasy, Gary Miller, Amir Nayyeri and Noel Walkington, I gave the first nearly-linear time algorithm for solving wide classes of higher order Laplacians via a flow-based approach. These systems are based on 3D meshes, or partitions of 3D space into simplices, and can be viewed as generalizations of edge-vertex incidence structures to include faces and simplices.

Randomized Numerical Linear Algebra

Randomized sampling is a widely used technique for reducing the sizes of problem instances. The ubiquity of large matrices in modern applications led to the popularity of algorithms that sample smaller matrix approximations that preserve various distance measures. Preserving the $\ell_2$-norm and $\ell_1$-norm of matrix-vector products allows one to approximately solve least squares and least absolute deviations problems, respectively, on the smaller sample. The $\ell_2$-norm is closely related to spectral sparsification, a key tool in the Laplacian paradigm, while the $\ell_1$-norm is of particular interest due to its sparsity-inducing behavior.

Study of randomized numerical linear algebraic algorithms led to routines that run in input sparsity time, which is linear in the matrix size plus a term depending on the intrinsic dimension. As intrinsic dimensions such as rank are often much smaller than the size of the data, these routines offer significant speedups. Size reductions in these routines are often obtained via oblivious projections generated independently from the input matrix. With Mu Li and Gary Miller, I pioneered the theoretical study of adaptive algorithms for subsampling matrices, leading to the current best theoretical guarantees for row sampling matrices. All matrices that arise during the computations
of these adaptive methods consist of subsets of rows of the original matrix, allowing the incorporation of algorithms designed for structured instances for further speedups.

Michael Cohen and I gave input-sparsity time algorithms for producing $\ell_1$-norm preserving samples whose sizes are nearly-linear in the intrinsic dimension. One formal parameterization of our result is that for any $n \times d$ matrix $A$, we generate in input sparsity time a matrix $A'$ consisting of $O(d \log d)$ rescaled rows of $A$ so that for all vectors $x$, $\|Ax\|_1$ and $\|A'x\|_1$ are within constant factors of each other. This significantly improves over previous results that give $A'$ with about $d^{2.5}$ rows, and circumvents lower bounds for oblivious projection methods in the $\ell_1$-setting. It relies on the first polynomial time algorithms for computing the Lewis change of density, a key tool in functional analysis that was viewed as cost-prohibitive to compute. We also gave an elementary proof of the convergence guarantees of this sampling process using techniques that originated from spectral sparsification of matrices. This proof simplifies and generalizes results by Talagrand on the embeddability of $\ell_1$-spaces.

**Combinatorial Graph Algorithms**

The scientific computing view of error as an integral part of computation is a valuable perspective for the study of efficient approximations to combinatorial problems. My collaborators and I have applied this perspective to data structures, parallel algorithms, and graph partitioning.

With Manoj Gupta, I gave the first sublinear work data structure for maintaining approximate weighted matchings in fully dynamic graphs. Key to our result is the maintenance of invariants in an approximate, error-carrying manner.

With Guy Blelloch and Kanat Tangwongsan, I gave a weighted, error-carrying variant of Luby’s algorithm for computing maximal independent sets. It led to work-efficient parallelizations for graph approximation algorithms such as set cover, max cover, and metric facility location.

With Gary Miller and Shen Chen Xu, I gave the first linear work, polylog depth routine for partitioning a graph into low diameter pieces. Several subsequent works joint with Cohen, Miller, Pachocki, Xu and by Blelloch, Dhulipala, and Shun demonstrated its usefulness as a building block for partitioning-based graph algorithms. These results led to the current best parallel routines for weighted spanners and approximate undirected shortest paths.